Hybridization in Genetic Algorithm with Hill Climbing to Optimize the Knapsack Problem

Amita Sharma[1] Kiran Jain[2]
[1]Research Scholar [2]Assistant Professor
[1,2]Department of Computer Science Engineering

Abstract— Genetic Algorithm (GA) is used to solve optimization problems. Knapsack is an optimization problem. To solve the knapsack problem we used two algorithms and their results were compared. To solve the knapsack problem we use genetic algorithm operators as selection, crossover, and mutation after that we perform the hybridization of GA and the results of both algorithms were compared. Memetic algorithm is combination of genetic algorithm and hill climbing on the Knapsack problem to obtain the more optimized result.

Key words: Hill Climbing, Knapsack Problem

I. Introduction

Genetic algorithms are adaptive algorithms proposed by John Holland in 1975 [1] and were described as heuristic search algorithms [2] based on the evolutionary ideas of natural selection and natural genetics by David Goldberg. These are powerful optimization techniques that employ concepts of evolutionary biology to evolve optimal solutions of a given problem. Genetic algorithm (GA) is mainly composed of two processes. First process is the selection of chromosomes from the current population for the production of next generation and the second process is manipulating the selected individuals by crossover and mutation techniques. The main work of selection operation is to determine which individual chromosomes are chosen for reproduction.

The main principle of selection is "the better is an individual; the higher is its chance of being parent." [1] [2] Selection reduces the search area by discarding the poor solutions. Crossover and mutation explore the search space for new promising solutions. The power of genetic algorithms comes from their ability to combine both exploration and exploitation in an optimal way. It is important better solutions go to the next generation more frequently than the poor solutions and exploration means poor solutions must have chance to go to next generation with the selection operation. Different selection strategies significantly affect performance of GA differently.

In practice, the population size in GA is finite and that affects its performance. Due to its limited population size, a genetic algorithm may also sample bad representatives of good search regions and good representatives of bad regions. A local search method can ensure fair representation of the different search areas by sampling their local optima which in turn can reduce the possibility of premature convergence. Incorporating a local search method can introduce new genes which can help to combat the genetic drift problem [3] [4] caused by the accumulation of stochastic errors due to finite populations.

In addition, a finite population can cause a genetic algorithm to produce solutions of less optimized solution as compared with the solution that can be produced using local search methods due to the difficulty of finding the best solution in the best found region for the genetic algorithm operators [5]. A local search method can introduce new genes which can help to combat the genetic drift problem [3, 4], caused by the accumulation of stochastic errors due to finite populations. It can also move the search towards the global optimum [6] which in turn can guarantee that the convergence rate is large enough to obstruct any genetic drift.

A local search method within a genetic algorithm can improve the exploiting ability of the search algorithm without affecting its exploring ability. If the balance between global exploration and local exploitation capabilities can be achieved, the algorithm can easily produce solutions of high accuracy [7].

In this study we will use the roulette wheel selection to generate new individuals and after that we will apply the two point crossover to produce new offspring, and then invert mutation to the new offspring produced by crossover operator with a mutation probability. Knapsack is a combinatorial optimization problem. It can be practically used in many situations such as capital budgeting, project scheduling, and selection of portfolio investments. In the portfolio investment selection, we will have many types of investment programs. Each programs offering different returns. In this case, the investor wants to find the sum of the selected values that is maximized under the constraint of limited funds for investment. Given a set of items, each with a mass and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items. The Knapsack problem is well–known NP-complete problem. Many methods have been used to find the optimal solution. Several Strategies for solving the knapsack problem are known, one of which is Branch and bound approach, Dynamic Programming and knapsack with multiprogramming is used.

A. *Unbounded Knapsack Problem (UKP):*

It places no upper bound on the number of copies of each kind of item and can be formulated as above except for that the only restriction on x is that it is a non-negative integer. Mathematically the unbounded knapsack problem can be formulated as in (1.1):

$$Maximize \sum_{i=1}^{n} v_i x_i, subject\ to \sum_{i=1}^{n} w_i x_i \le W, x_i \ge 0 \tag{1.1}$$

The 0-1 knapsack problem restricts the number x_i of copies of each kind of item to zero or one. Mathematically the 0-1-knapsack problem can be formulated as in (1.2):

Let there be n items, z_1 to z_n where z_i has a value v_i and weight w_i. x_i is the number of copies of the item z_i, which, mentioned above, must be zero or one. The maximum weight that we can carry in the bag is *W*. It is common to assume that all values and weights are nonnegative. To simplify the representation, we also assume that the items are listed in increasing order of weight.

$$Maximize \sum_{i=1}^{n} v_i x_i, subject\ to \sum_{i=1}^{n} w_i x_i \leq W, x_i \in \{0,1\} \qquad (1.2)$$

Maximize the sum of the values of the items in the knapsack so that the sum of the weights must be less than or equal to the knapsack's capacity.

The remainder of this paper is organized as follows: Section II presents a literature review on selection strategy. Section III contains an introduction to hill climbing local search. Section IV discusses the proposed algorithm and section V contains the experimental results. Lastly, Section VI contains the conclusion.

II. LITERATURE REVIEW

Several researchers have studied the performance of GA using different selection strategies. The performance is usually evaluated in terms of convergence rate and number of generations to reach the optimal solution. Rakesh Kumar et al. proposed a novel crossover operator that uses the principle of Tabu search. They compared the proposed crossover with PMX and found that the proposed crossover yielded better results than PMX [8]. Sanusi used the two Evolutionary Algorithm techniques, that is, Genetic Algorithm and Memetic Algorithm have been applied to solve knapsack problem. Memetic algorithm converges the faster than genetic algorithm even as it produce more optimal result [9]. Pisinger [10] gave an overview of all recent exact solution approaches for the Knapsack problem, and showed that the Knapsack problem is still difficult for these algorithms to solve. However Pisinger's book does not offer any genetic algorithms for solving the Knapsack problem. Maya Hristakeva use the genetic algorithm to solve the knapsack problem ,in this paper she implements two selection functions that is roulette wheel and group selection and the result from these two are different depending upon the usage of elitism used or not. The result of program shows that implementation of good selection method and elitism is very important [11].

Bjornsdotter et al. [12] proposed a memetic algorithm for feature selection in volumetric data containing spatially distributed clusters of informative features in neuroscience application. They concluded that the proposed MA identified a majority of relevant features as compared to genetic algorithm. Sivaraj et al. [13] discussed about a novel approach to improve the performance of genetic algorithm by using selective initialization, which aims at supplying more fit individuals in the beginning. The result shows that the selective initialization enhances the convergence velocity and produces more optimal solution than existing schemes used in generic genetic algorithm.

III. HILL CLIMBING

Hill Climbing algorithm searches for a better solution in the neighborhood. If it finds a better solution, it changes the current solution with this new one. If the new solution is not the better one then the algorithm stops and keeps the current local optimum solution. The simplex method of linear programming is also a hill climbing procedure that moves from one extreme point solution to another, using an exact neighborhood.

Algorithm Hill Climbing (Iterative improvement)
begin
    i:=initial solution
    repeat
        generate an *s* ϵ *Neighbour(i);*
        if fitness(*s*) > fitness(*i*) then *i*:=*s*;
    until f(*s*) ≤ f(*i*) for all s ϵ *Neighbour(i);*
end

Hill Climbing algorithm suffers from the problem of local optima. In the hill climbing algorithm, we find that either the move is good or bad. If the move is good then we will set that point as a current point and otherwise use the same.

IV. PROPOSED ALGORITHM

In this section, a hybrid genetic algorithm is described in which hill climbing local search is applied on the initial randomly generated population. In this algorithms, De Jong's guidelines, which is to start with a relatively high crossover probability (Pc ≥ 0.6), relatively low mutation probability (Pm, 0.001-0.1), and a moderately sized population is used. The selections of parameter values are very depend on the problem to be solved [14, 15].

In the simple genetic algorithm, following steps are followed:

1) Initialization (using binary encoding).
2) Roulette Wheel Selection (that selects only two individuals at a time that will take part in reproduction after that).
3) Two Point Crossover (two crossover points are chosen and content between these points are exchanged between two mated parents to produce two offsprings).
4) Invert Mutation (Two bits are chosen randomly and in between substring are inverted).

In the proposed hybrid algorithm, following steps are followed:

1) Initialization (using binary encoding).
2) Hill climbing applied on the population generated after initialization.
3) Roulette Wheel Selection (that selects only two individuals from the population got after applying local search).
4) Two Point Crossover.
5) Invert Mutation (Two bits are chosen randomly and in between substring is inverted).

For both the algorithms in this study, termination is performed when number of generation reached the maximum number of generation. The maximum number of generation is entered at runtime of program. The size of population is also entered at runtime of program.

V. EXPERIMENTAL RESULT

In this paper, MATLAB code is developed for genetic algorithm. The problem considered is the 0-1 Knapsack Problem. Knapsack problem is one of the NP hard problems often used as a benchmark for optimization techniques. Knapsack has several applications like planning, logistics, manufacture of microchips and DNA sequencing. Knapsack problem is to find the optimal solution to fill the knapsack profit. Likewise, we apply the same technique to obtain the optimal solution for eating meal. So, we can get maximum calories by using the optimal solution.

The following parameters are used in this implementation:-

- Number of Items in the Knapsack Problem.
- Number of Individuals
- Weight Of Items
- Value of items
- Knapsack capacity
- Number of Generations

The result when using different dataset for Knapsack Problem are shown below:-

1) Number of items in Knapsack problem 28
   Number of individuals:-50
   Weight of Items w = [30, 20, 125, 5, 80, 25, 35, 73, 12, 15, 15, 40, 5, 10, 10, 12, 10, 9, 0, 20, 60, 40, 50, 36, 49, 40, 19, 150];
   Value of items v= [1898, 440, 22507, 270, 14148, 3100, 4650, 30800, 615, 4975, 1160, 4225, 510, 11880, 479, 440, 490, 330, 110, 560, 24355, 2885, 11448, 4550, 750, 3720, 1950, 10500];
   Knapsack Capacity = 600
   Number of generations =100
2) Number of items in Knapsack problem 28
   Number of individuals:-50
   Weight of Items w= [45, 0, 85, 150, 65, 95, 30, 0, 170, 0, 40, 25, 20, 0, 0, 25, 0, 0, 25, 0, 165, 0, 85, 0, 0, 0, 0, 100];
   Value of items v= [1898, 440, 22507, 270, 14148, 3100, 4650, 30800, 615, 4975, 1160, 4225, 510, 11880, 479, 440, 490, 330, 110, 560, 24355, 2885, 11448, 4550, 750, 3720, 1950, 10500];
   Knapsack Capacity= 600
   Number of generations =100
3) Number of items in Knapsack problem 60
   Number of individuals:-100
   Weight of Item w = [47, 774, 76, 56, 59, 22, 42, 1, 21, 760, 818, 62, 42, 36, 785, 29, 662, 49, 608, 116, 834, 57, 42, 39, 994, 690, 27, 524, 23, 96, 667, 490, 805, 46, 19, 26, 97, 71, 699, 465, 53, 26, 123, 20, 25, 450, 22, 979, 75, 96, 27, 41, 21, 81, 15, 76, 97, 646, 898, 37];
   Value of Item v=[2, 77, 6, 67, 930, 3, 6, 270, 33, 13, 110, 21, 56, 974, 47, 734, 238, 75, 200, 51, 47, 63, 7, 6, 468, 72, 95, 82, 91,83, 27, 13, 6, 76, 55, 72, 300, 6, 65, 39, 63, 61, 52, 85, 29, 640, 558, 53, 47, 25, 3, 6, 568, 6, 2, 780, 69, 31, 774, 22];
   Knapsack Capacity=6000
   Number of Generations=150

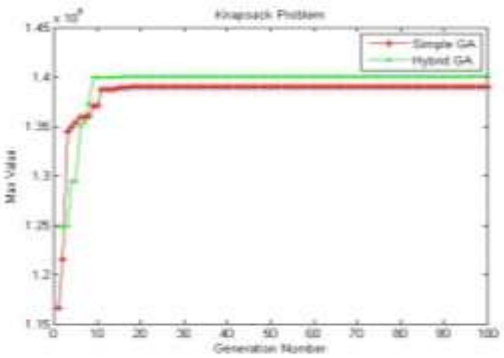

Fig. 1: Comparison of maximum values of Knapsack for data set 1

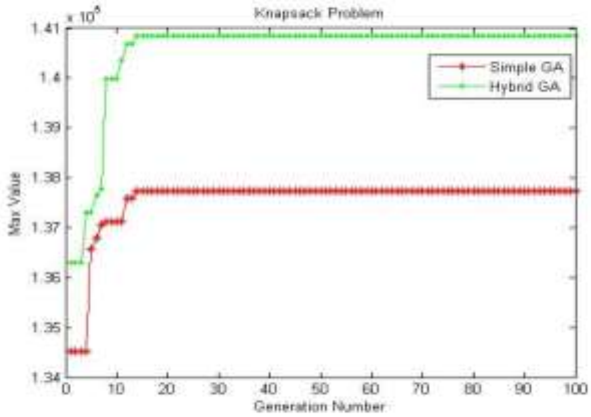

Fig. 2: Comparison of maximum values of Knapsack for data set 2

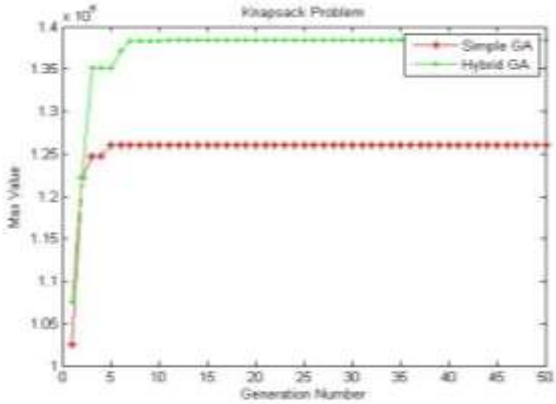

Fig. 3: Comparison of maximum values of Knapsack for data set 3

VI. CONCLUSION

In this paper, hybridization of genetic algorithm with hill climbing is proposed for Knapsack Problem. Local search is applied on the initialization stage of genetic algorithm considering that if we start with a good population then convergence speed will be increased and more optimal result can be found. The proposed algorithm is tested on three data set of benchmark 0-1 Knapsack optimization problem. It is found that as we start with better chromosomes, more optimal result is found by proposed algorithm than simple genetic algorithm.

REFERENCES

[1] J. Holland, Adaptation in natural and artificial systems, Ann Arbor: University of Michigan Press, 1975.
[2] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison Wesley Longman Inc., 1989.
[3] H. Asoh and H. Muhlenbein, "On the mean convergence time of evolutionary algorithms without selection and mutation," in Parallel Problem Solving from Nature, Berlin, Springer-Verlag, 1994, pp. 88-97.
[4] D. Thierens, D. Goldberg and P. Guimaraes, "Domino Convergence, Drift, and the temporal-salience structure of problems," in IEEE International Conference on Evolutionary Computation Anchorage, USA, 1998.

[5] C. Reeves, "Genetic algorithms and neighbourhood search," AISB Workshop in Evolutionary Computation, Springer, vol. 865, pp. 115-130, 1994.
[6] W. E. Hart, "Adaptive Global Optimization with local search," University of California, San Diego, 1994.
[7] F. G. Lobo and D. E. Goldberg, "Decision Making in a Hybrid Genetic Algorithm," in IEEE International Conference on Evolutionary Computation, USA, 1997.
[8] R. Kumar and Jyotishree, "Novel Knowledge Based Tabu Crossover in Genetic Algorithm," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 8, pp. 78-82, 2012.
[9] H. A. Sanusi, A. Zubair and R. O. Oladele, "Comparative Assessment of Genetic and Memetic Algorithms," Journal of Emerging Trends in Computing and Information Science, vol. 2, no. 10, pp. 498-508, 2011.
[10] D. Pisinger, "Where are the Hard Knapsack problems?," Computers and Operation Research, vol. 32, no. 9, pp. 2271-2284, 2005.
[11] M. Hristakeva and D. Shrestha, "Solving 0-1 Knapsack Problem using Genetic Algorithm," in MICS 2004 Proceedings, 2004.
[12] M. Bjornsdotter and J. Wessberg, "A memetic algorithm for selection of 3D clustered featured with applications in neuroscience," in Internatioonal Conference on Pattern Recognition, IEEE, 2010.
[13] R. Sivaraj, T. Ravichandran and R. Devipriya, "Boosting Performance of Genetic Algorithm through selective Initialization," European Journal of Scientific Research, vol. 68, no. 1, pp. 93-100, 2012.
[14] K. A. De Jong and W. M. Spears, "Using Genetic Algorithm to solve NP Complete Problems," in Proceedings of third International Conference on Genetic Algorithm, Morgan Kaufman, 1989.
[15] K. A. De Jong, "An Analysis of the behavior of a class of Genetic Adaptive Systems," University of Michigan, Ann Arbor, 1975.

